

We claim:

1. A processor with an instruction class controllable pipeline comprising:

a program storage unit holding a diverse plurality of class one and class two executable function instructions, the class one instructions having a shorter execution latency and
5 the class two instructions having a longer execution latency;

a fetch stage for fetching an instruction from the program storage unit to be stored in an instruction register;

a decode stage for classifying and decoding the instruction stored in the instruction register and generating an instruction class indication and storing decoded
10 instructions in a decode register;

an adaptable pipeline control unit responsive to the instruction class indication for adapting the pipeline to the instruction class; and

an adaptable execution stage for execution of a decoded instruction stored in the decode register, the decoded instruction being a class one instruction or a class two instruction.

15 2. The processor of claim 1, wherein the fetch stage further comprises:

a program counter and an instruction memory fetch mechanism which are operable to begin instruction processing by fetching one or more instructions from the program storage unit.

3. The processor of claim 1, wherein the executable function instructions comprise:

20 additions, subtractions, multiplications, divisions, compares, ANDs, ORs, ExclusiveORs, NOTs, shifts, rotates, permutes, bit operations, moves, loads, stores, communications and variations and combinations thereof.

4. The processor of claim 1, wherein the adaptable pipeline control unit further comprises:

a pipeline control mechanism for class one instructions to execute in a first time period; and

5 a pipeline control mechanism for class two instructions to execute in a second longer time period.

5. The processor of claim 4 wherein the pipeline control mechanism for class one instructions further comprises:

10 control for normal pipeline sequencing for class one instructions to execute in a first time period.

6. The processor of claim 4 wherein the pipeline control mechanism for class two instructions further comprises:

an instruction register feedback multiplexer;

a decode register feedback multiplexer;

15 a program counter and program counter update function;

a hold instruction register signal to control the instruction register feedback multiplexer to hold the contents of the instruction register for a second longer time period upon detection of a class two instruction;

20 a hold decode register signal to control the decode register feedback multiplexer to hold the contents of the decode register for a second longer time period upon detection of a class two instruction;

a hold program counter signal to control the program counter update function to hold the contents of the program counter for a second longer time period upon detection of a class two instruction; and

5 a control for extending pipeline sequencing for class two instruction to execute in a second longer time period.

7. The processor of claim 4 wherein the pipeline control mechanism for class two instructions further comprises:

an instruction register gated clock;

a decode register gated clock;

10 a program counter gated clock;

instruction register clock gating logic responsive to the instruction class indication to extend the instruction register gated clock for a second longer time period upon detection of a class two instruction;

15 decode register clock gating logic responsive to the instruction class indication to extend the decode register gated clock for a second longer time period upon detection of a class two instruction;

program counter clock gating logic responsive to the instruction class indication to extend the program counter gated clock for a second longer time period upon detection of a class two instruction; and

20 control for extending pipeline sequencing for class two instructions to execute in a second longer time period.

8. The processor of claim 7 wherein the instruction register gated clock, the decode register gated clock, and the program counter gated clock are a single gated clock.

9. The processor of claim 1 wherein the adaptable execution stage further comprises:

a class one execution unit operable to execute a class one instruction stored in the decode register; and

5 a class two execution unit operable to execute a class two instruction stored in the decode register.

10. The processor of claim 1 wherein the decode stage further operates to decode an opcode field to classify an instruction.

11. The processor of claim 1 wherein the decode stage further operates to decode an opcode field and decode of a data type field to classify an instruction.

10 12. The processor of claim 4 wherein the adaptable pipeline control unit further comprises:

a programmable clock gating mode indicator that specifies a normal clock gating mode and a slow down clock gating mode; and

15 control for extending pipeline sequencing both class 1 instructions and class 2 instructions to execute in a third longer time period when the programmable clock gating mode indicator specifies a slow down clock gating mode.

13. A method for processor performance and power optimization of an instruction class adaptable pipeline processor supporting at least two classes of instructions with a first class operable at a high frequency and a second class operable at a lower frequency and where the
20 instructions operable at a higher frequency class can be specified to operate at the higher frequency and the lower frequency and where the instructions operable at the lower frequency can be specified to only operate at the lower frequency class, the method comprising:

programming the instruction class adaptable pipeline processor creating an application program containing a mix of two classes of instructions to meet functional requirements with all instructions used in the program operable at the high frequency class specified as class 1 instructions; and

5 modifying the application program to meet performance and power requirements of an application by changing, where appropriate, class 1 instructions to class 2 instructions.

14. A method for processor performance and power optimization of claim 13 wherein modifying the application program to meet performance and power requirements of an application, the method further comprises:

10 appropriately programming a programmable clock gating mode to cause a specifiable majority of the instructions of the class adaptable pipeline processor to execute at a lower clock frequency than the class 2 clock frequency, to meet performance and power requirements of an application.

15 15. A very long instruction word (VLIW) processor with a plurality of instruction class controllable pipelines comprising:

 a VLIW storage unit holding a diverse plurality of class one and class two executable function instructions located in multiple instruction slot format VLIWs, the class one instructions having a shorter execution latency and the class two instructions having a longer execution latency;

20 a VLIW fetch stage for fetching a VLIW from a VLIW storage unit to be stored in a VLIW instruction register (VIR);

 a decode stage for classifying and decoding the plurality of executable function instructions stored in the VIR, and generating an instruction class indication for each of the

plurality of executable function instructions and storing decoded instructions in a plurality of instruction slot specific decode registers;

an adaptable pipeline control unit responsive to the instruction class indications from the classified plurality of executable function instructions for adapting the pipeline to the instruction class; and

a plurality of adaptable execution stages each operable for execution of a decoded instruction stored in an instruction slot specific decode register, the decoded instruction being a class one instruction or a class two instruction.

16. The VLIW processor of claim 15, wherein the VLIW fetch stage further comprises:

a VLIW memory control unit which is operable to begin VLIW processing by fetching a VLIW from the VLIW storage unit.

17. The processor of claim 15, wherein the executable function instructions comprise:

additions, subtractions, multiplications, divisions, compares, ANDs, ORs, ExclusiveORs, NOTs, shifts, rotates, permutes, bit operations, moves, loads, stores,

communications and variations and combinations thereof;

18. The processor of claim 15, wherein the adaptable pipeline control unit further comprises:

a pipeline control mechanism for a VLIW, consisting of all class one instructions, to execute in a first time period; and

a pipeline control mechanism for a VLIW, consisting of at least one class two instruction, to execute in a second longer time period.

19. The processor of claim 18 wherein the pipeline control mechanism for a VLIW consisting of all class one instructions further comprises:

control for normal pipeline sequencing for the class one instructions to execute in a first time period.

20. The processor of claim 18 wherein the pipeline control mechanism for a VLIW consisting of at least one class two instruction further comprises:

- 5 VIR state maintaining multiplexers;
- decode register state maintaining multiplexers;
- a program counter and program counter update function;
- a hold VIR signal to control the VIR state maintaining multiplexers to hold the contents of the VIR for a second longer time period upon detection of at least one class two
- 10 instruction;
- a plurality of hold decode register signals to control the decode register state maintaining multiplexers to hold the contents of the decode registers for a second longer time period upon detection of at least one class two instruction;
- a hold program counter signal to control the program counter update function to
- 15 hold the contents of the program counter for a second longer time period upon detection of at least one class two instruction; and
- control for extending pipeline sequencing for the VLIW to execute in a second longer time period.

21. The processor of claim 18 wherein the pipeline control mechanism for a VLIW consisting of at least one class two instruction further comprises:

- 20 a VIR gated clock;
- a plurality of decode register gated clocks;
- a program counter gated clock;

VIR clock gating logic responsive to the plurality of instruction class indications to extend the VIR gated clock for a second longer time period upon detection of at least one class two instruction;

5 a plurality of decode register clock gating logic responsive to the plurality of instruction class indications to extend the decode register gated clocks for a second longer time period upon detection of at least one class two instruction;

program counter clock gating logic responsive to the plurality of instruction class indications to extend the program counter gated clock for a second longer time period upon detection of at least one class two instruction; and

10 control for extending pipeline sequencing for the VLIW to execute in a second longer time period.

22. The processor of claim 15 wherein each adaptable execution stage of the plurality of adaptable execution stages further comprises:

15 a class one execution unit operable to execute a class one instruction stored in the decode register; and

a class two execution unit operable to execute a class two instruction stored in the decode register.

23. A method for synthesis of an adaptable logic function consisting of a first class logic sub-function with a first worst case timing path length meeting a first maximum clock frequency
20 and a second class logic sub-function with a second worst case timing path length meeting a second maximum clock frequency where the first worst case timing path length is less than the second worst case timing path length, the method comprising:

creating a non-inverting hardware description language (HDL) module (Module1) with a small propagation delay parameter and its associated model for the first class logic sub-function;

5 creating a non-inverting HDL module (Module2) with a small propagation delay parameter and its associated model for the second class logic sub-function;

replacing the small propagation delay parameter of Module1 with a propagation delay equal to the difference between the first worst case timing path length of the first class logic sub-function and the second worst case timing path length of the second class logic sub-function and accounting for the small propagation delay parameter of Module2;

10 instantiating Module1 in the first class logic sub-function execution path;

instantiating Module2 in the second class logic sub-function execution path; and

synthesizing the adaptable logic function for a plurality of timing paths with the period of the clock set to the second worst case timing path length of the second class logic sub-function and accounting for the small propagation delay parameter of Module2.

15 24. The method of claim 23 further comprising:

creating a timing view of Module1 and Module2 for a place and route tool based on the chosen timing parameters;

applying timing driven place and route techniques for the synthesized logic;

20 replacing the Module1 propagation delay parameter with a small propagation delay parameter;

timing the first class logic sub-function at the first maximum clock frequency and accounting for the small delay parameter of Module1; and

timing the second class logic sub-function at the second maximum clock frequency and accounting for the small delay parameter of Module2.

25. The method of claim 23 wherein the first class logic sub-function further comprises:

5 a grouping of sub-functions that all have a timing path less than or equal to the first worst case timing path length and operable at a first maximum clock frequency.

26. The method of claim 23 wherein the second class logic sub-function further comprises:

a grouping of sub-functions that all have a timing path less than or equal to the second worst case timing path length and operable at a second maximum clock frequency.

10 27. The method of claim 23 wherein the adaptable logic function is an adaptable execution unit of a processor supporting at least two classes of logic sub-functions.